CLAIMS:

1. A method for programming forwarding tables for switches for multipathing in a subnet of a switched fabric including at least a host system, a target system and switches each having one or more ports interconnected via links, said method comprising:

determining all possible links between all ports on the subnet during topology discovery;

creating an all port connectivity table which records all port-to-port connectivity information;

creating an all switch shortest paths table which records all the shortest paths between every switch pair on the subnet based the port-to-port connectivity information; and

computing forwarding tables for respective switches on the subnet that allow usage of multiple paths between switch pairs based on the port-to-port connectivity information and based on the shortest paths between every switch pair.

2. The method as claimed in claim 1, further comprising:

downloading the forwarding tables to respective switches on the subnet that allow usage of multiple paths between switch pairs; and

enabling respective switches on the subnet to route data packets from the host system to the target system via mutiple paths through the switched fabric.

3. The method as claimed in claim 1, wherein each of said host system and said target system includes a channel adapter (CA) installed supporting one or more ports with each port having multiple local identifiers (LIDs) assigned thereto for multipathing.

4. The method as claimed in claim 3, wherein each port on the subnet supports a unique 16-bit LID and a LID Mask Control (LMC) which specifies the number of low order bits of the LID to mask when checking a received destination LID against the port's destination LID.

5. The method as claimed in claim 1, wherein said forwarding tables are computed to ensure loop-less paths and allow ports to be addressed by multiple local identifiers (LIDs), and wherein said all-port connectivity and all-switch shortest paths tables are constantly updated reflecting any dynamic changes to the subnet topology.

6. The method as claimed in claim 1, wherein said forwarding tables are computed based on the principle that only the shortest path between a given switch pair is guaranteed to overlap with other shortest paths that either originate from or destined to some intermediate port that exists on the shortest path between the original switch pair.

7. The method as claimed in claim 1, wherein a forwarding table for a switch is computed by:

determining a destination switch to which a destination port is directly connected,

identifying all the links that exist between the destination switch and other switches in the subnet;

sorting all the links by respective originating port number in an ascending order;

picking an appropriate link and identifying the switch to which the link is connected at the other end;

determining the best route between the switch identified and the switch for which the forwarding table is being constructed; and

inputting associated outport number at a designated location in the forwarding table.

8. The method as claimed in claim 1, wherein said shortest paths between every switch pair are computed utilizing an All Pair Shortest Paths (APSP) algorithm, and each shortest path from the source to the destination switch is represented by a <Port, Cost> duple in which port is the port number of the source switch where the path originates and cost is the path cost metric that is computed based on a hop count, a message transfer (MTU) size, a link speed, width and other port and link characteristics.

9. The method as claimed in claim 4, wherein a forwarding table for a switch is computed by:

determining a destination switch to which a destination port is directly connected,

identifying all the links that exist between the destination switch and other switches in the subnet;

sorting all the links by respective originating port number in an ascending order;

picking an appropriate link and identifying the switch to which the link is connected at the other end;

determining the best route between the switch identified and the switch for which the forwarding table is being constructed; and

inputting associated outport number at a designated location in the forwarding table.

10. The method as claimed in claim 4, wherein each of said forwarding tables for switches in the subnet is computed, when multiple LIDs are assigned to channel adapter (CA) ports, by a multipath assignment algorithm configured to:

receive information during the topology discovery including the number of multiple paths (m) for configuration, the switch LID for which the forwarding table is being built (LID_S), and the LID in the forwarding table which is currently identifying the correct outport (LID_d) for forwarding data packets;

determine the base LID (LID_{dbase}) for the given LID_d and determine if the base LID (LID_{dbase}) for the given LID_d corresponds to a switch using the all port connectivity table;

if the base LID (LID_{dbase}) for the given LID_d corresponds to a switch using the all port connectivity table, check if the base LID (LID_{dbase}) for the given LID_d corresponds to the switch

LID for which the forwarding table is being built (LID_S);

if the base LID (LID_{dbase}) for the given LID_d corresponds to the switch LID for which the forwarding table is being built (LID_S), indicate that there is no route for the given LID_d;

if the base LID (LID_{dbase}) for the given LID_d happens to be any switch other than the switch LID for which the forwarding table is being built (LID_S), check if the base LID (LID_{dbase}) for the given LID_d corresponds to the given LID_d;

if the base LID (LID_{dbase}) for the given LID_d corresponds to the given LID_d, set an arbitrary LIDx as the base LID (LID_{dbase}) for the given LID_d, and proceed to get the port number of switch LID_S for the best route between switch LID_S and LIDx from the all switch shortest paths table;

if the base LID (LID_{dbase}) for the given LID_d does not correspond to a switch using the all port connectivity table, identify the peer node and port to which the port with LID_{dbase} is connected from the all port connectivity table and determine if the peer node is a switch;

if the peer node does not correspond to a switch, indicate that there is no route for the given LID_d;

if the peer node corresponds to a switch, determine if the peer switch LID (LID_{sdest}) corresponds to the switch LID for which the forwarding table is being built (LID_S);

if the peer switch LID (LID_{sdest}) corresponds to LID_S, determine that the switch LID_S and port LID_{dbase} are directly connected, and get the appropriate port number of switch LID_S from the all port connectivity table;

if the peer switch LID (LID_{sdest}) does not correspond to LID_S, identify all the links that are directly connected to other switches from the peer switch LID (LID_{sdest}) and determine the number of (N) links that connect switch LID_{sdest} where N is greater than "0";

if the number of (N) links is not greater than "0", indicate that there is no route for the given LID_d;

if the number of (N) links is greater than "0", set the offset (n) of LID_d from LID_{dbase}, and determine if the number of multipaths (m) is less than the offset (n) of LID_d from LID_{dbase} and if the number of (N) links that connect switch LID_{sdest} to other switch ports is less than the offset (n) of LID_d from LID_{dbase};

if either the multipaths (m) or the (N) links is less than the offset (n) of LID_d from LID_{dbase}, indicate that there is no route for the given LID_d;

if neither the multipaths (m) nor the (N) links is less than the offset (n) of LID_d from LID_{dbase}, store the LIDs of switches to which N links are connected in a list O(i) and sort the list by the port number of switch LID_{sdest} from where each of N links originate in an ascending order;

set the list O(i).LID to the LID of the switch that the link O(n) is connected at the other end, and check if O(i).LID corresponds to the switch LID for which the forwarding table is being built (LID_S);

if the list O(i).LID corresponds to the switch LID for which the forwarding table is being built (LID_S), set an arbitrary LIDx as the LID of the switch to which the port LID_d is directly connected (LID_{sdest}) and then get the port number of switch LID_S for the best route between

switch LID_S and LIDx from the all switch shortest paths table;

if the list O(i).LID does not correspond to the switch LID for which the forwarding table is being built (LID_S), set an arbitrary LIDx as the O(n).LID, and then get the port number of switch LID_S for the best route between switch LID_S and LIDx from the all switch shortest paths table; and

determine if the LID_d is the last LID assigned to a port or switch in the subnet, and terminate computation of the forwarding table when LID_d is the last LID assigned to a port or switch in the subnet.

11. A data network, comprising:

a host system having at least one channel adapter (CA) installed therein supporting one or more ports with each port having multiple local identifiers (LIDs) assigned thereto for multipathing;

at least one target system having at least one channel adapter (CA) installed therein supporting one or more ports with each port having multiple local identifiers (LIDs) assigned thereto for multipathing;

a switched fabric comprising a plurality of different switches which interconnect said host system via CA ports to said remote system via CA port along different physical links for data communications; and

a fabric manager provided in said host system for making topology discovery, assigning

local identifiers (LIDs) to all ports that are connected in the switched fabric, and programming forwarding tables for switches in the switched fabric, wherein said fabric manager programs forwarding tables for switches for multipathing by:

determining all possible links between all ports that are connected in the switched fabric during topology discovery;

creating an all port connectivity table which records all port-to-port connectivity information;

creating an all switch shortest paths table which records all the shortest paths between every switch pair on the switched fabric based the port-to-port connectivity information; and

computing forwarding tables for respective switches on the switched fabric that allow usage of multiple paths between switch pairs based on the port-to-port connectivity information and based on the shortest paths between every switch pair.

12. The data network as claimed in claim 11, wherein said fabric manager is configured to download the forwarding tables to respective switches on the switched fabric that allow usage of multiple paths between every switch pair, and enable respective switches on the switched fabric to route data packets from the host system to the target system via multiple paths through the switched fabric.

13. The data network as claimed in claim 11, wherein each port on the subnet supports a unique 16-bit LID and a LID Mask Control (LMC) which specifies the number of low order bits of the LID to mask when checking a received destination LID against the port's destination LID.

14. The data network as claimed in claim 11, wherein said forwarding tables are computed to ensure loop-less paths and allow ports to be addressed by multiple local identifiers (LIDs), and wherein said all-port connectivity and all-switch shortest paths tables are constantly updated reflecting any dynamic changes to the subnet topology.

15. The data network as claimed in claim 11, wherein said forwarding tables are computed based on the principle that only the shortest path between a given switch pair is guaranteed to overlap with other shortest paths that either originate from or destined to some intermediate port that exists on the shortest path between the original switch pair.

16. The data network as claimed in claim 11, wherein said fabric manager is configured to compute a forwarding table for a switch by:

determining a destination switch to which a destination port is directly connected,

identifying all the links that exist between the destination switch and other switches in the subnet;

sorting all the links by respective originating port number in an ascending order;

picking an appropriate link and identifying the switch to which the link is connected at the other end;

determining the best route between the switch identified and the switch for which the forwarding table is being constructed; and

inputting associated outport number at a designated location in the forwarding table.

17. The data network as claimed in claim 11, wherein said shortest paths between every port pair are computed utilizing an All Pair Shortest Paths (APSP) algorithm, and each shortest path from the source to the destination switch is represented by a <Port, Cost> duple in which port is the port number of the source switch where the path originates and cost is the path cost metric that is computed based on a hop count, a message transfer (MTU) size, a link speed, width and other port and link characteristics.

18. The data network as claimed in claim 11, wherein said fabric manager is provided with an algorithm for computing forwarding tables for switches, when multiple LIDs are assigned to channel adapter (CA) ports, by:

receiving information during the topology discovery including the number of multiple paths (m) for configuration, the switch LID for which the forwarding table is being built (LID_S), and the LID in the forwarding table which is currently identifying the correct outport (LID_d) for

forwarding data packets;

determining the base LID (LID_{dbase}) for the given LID_d and if the base LID (LID_{dbase}) for the given LID_d corresponds to a switch using the all port connectivity table;

if the base LID (LID_{dbase}) for the given LID_d corresponds to a switch using the all port connectivity table, checking if the base LID (LID_{dbase}) for the given LID_d corresponds to the switch LID for which the forwarding table is being built (LID_S);

if the base LID (LID_{dbase}) for the given LID_d corresponds to the switch LID for which the forwarding table is being built (LID_S), indicating that there is no route for the given LID_d;

if the base LID (LID_{dbase}) for the given LID_d happens to be any switch other than the switch LID for which the forwarding table is being built (LID_S), checking if the base LID (LID_{dbase}) for the given LID_d corresponds to the given LID_d;

if the base LID (LID_{dbase}) for the given LID_d corresponds to the given LID_d, setting an arbitrary LIDx as the base LID (LID_{dbase}) for the given LID_d, and getting the port number of switch LID_S for the best route between switch LID_S and LIDx from the all switch shortest paths table;

if the base LID (LID_{dbase}) for the given LID_d does not correspond to a switch using the all port connectivity table, identifying the peer node and port to which the port with LID_{dbase} is connected from the all port connectivity table and determining if the peer node is a switch;

if the peer node does not correspond to a switch, indicating that there is no route for the given LID_d;

if the peer node corresponds to a switch, determining if the peer switch LID (LID_{sdest}) corresponds to the switch LID for which the forwarding table is being built (LID_S);

if the peer switch LID (LID_{sdest}) corresponds to LID_S, determining that the switch LID_S and port LID_{dbase} are directly connected, and getting the appropriate port number of switch LID_S from the all port connectivity table;

if the peer switch LID (LID_{sdest}) does not correspond to LID_S, identifying all the links that are directly connected to other switches from the peer switch LID (LID_{sdest}) and determining the number of (N) links that connect switch LID_{sdest} where N is greater than "0";

if the number of (N) links is not greater than "0", indicating that there is no route for the given LID_d;

if the number of (N) links is greater than "0", setting the offset (n) of LID_d from LID_{dbase}, and determining if the number of multipaths (m) is less than the offset (n) of LID_d from LID_{dbase} and if the number of (N) links that connect switch LID_{sdest} to other switch ports is less than the offset (n) of LID_d from LID_{dbase};

if either the multipaths (m) or the (N) links is less than the offset (n) of LID_d from LID_{dbase}, indicating that there is no route for the given LID_d;

if neither the multipaths (m) nor the (N) links is less than the offset (n) of LID_d from LID_{dbase}, storing the LIDs of switches to which N links are connected in a list O(i) and sorting the list by the port number of switch LID_{sdest} from where each of N links originate in an ascending order;

setting the list O(i).LID to the LID of the switch that the link O(n) is connected at the other end, and checking if O(i).LID corresponds to the switch LID for which the forwarding table is being built (LID_S);

if the list O(i).LID corresponds to the switch LID for which the forwarding table is being built (LID_S), setting an arbitrary LIDx as the LID of the switch to which the port LID_d is directly connected (LID_{sdest}) and then getting the port number of switch LID_S for the best route between switch LID_S and LIDx from the all switch shortest paths table;

if the list O(i).LID does not correspond to the switch LID for which the forwarding table is being built (LID_S), setting an arbitrary LIDx as the O(n).LID, and then getting the port number of switch LID_S for the best route between switch LID_S and LIDx from the all switch shortest paths table; and

determining if the LID_d is the last LID assigned to a port or switch in the subnet, and terminating computation of the forwarding table when LID_d is the last LID assigned to a port or switch in the subnet.

19. A computer readable medium comprising instructions that, when executed by a computer system, cause the computer system to:

determine all possible links between all ports on a subnet including at least a host system, a target system and switches each having one or more ports interconnected via links during topology discovery;

create an all port connectivity table which records all port-to-port connectivity information;

create an all switch shortest paths table which records all the shortest paths between every port pair on the subnet based the port-to-port connectivity information; and

compute forwarding tables for respective switches on the subnet that allow usage of multiple paths between port pairs based on the port-to-port connectivity information and based on the shortest paths between every port pairs.

20. The computer readable medium as claimed in claim 19, further causing the computer system to:

download the forwarding tables to respective switches on the subnet that allow usage of multiple paths between port pairs; and

enable respective switches on the subnet to route data packets from the host system to the target system via multiple paths through the switched fabric.

21. The computer readable medium as claimed in claim 19, wherein each port on the subnet supports a unique 16-bit LID and a LID Mask Control (LMC) which specifies the number of low order bits of the LID to mask when checking a received destination LID against the port's destination LID.

22. The computer readable medium as claimed in claim 19, wherein said forwarding tables are computed to ensure loop-less paths and allow ports to be addressed by multiple local identifiers (LIDs), and wherein said all-port connectivity and all-switch shortest paths tables are constantly updated reflecting any dynamic changes to the subnet topology.

23. The computer readable medium as claimed in claim 19, wherein said forwarding tables are computed based on the principle that only the shortest path between a given port pair is guaranteed to overlap with other shortest paths that either originate from or destined to some intermediate port that exists on the shortest path between the original port pair.

24. The computer readable medium as claimed in claim 19, wherein a forwarding table for a switch is computed by:

determining a destination switch to which a destination port is directly connected,

identifying all the links that exist between the destination switch and other switches in the subnet;

sorting all the links by respective originating port number in an ascending order;

picking an appropriate link and identifying the switch to which the link is connected at the other end;

determining the best route between the switch identified and the switch for which the forwarding table is being constructed; and

inputting associated outport number at a designated location in the forwarding table.

25. The computer readable medium as claimed in claim 19, wherein said shortest paths between every switch pair are computed utilizing an All Pair Shortest Paths (APSP) algorithm, and each shortest path from the source to the destination switch is represented by a <Port, Cost> duple in which port is the port number of the source switch where the path originates and cost is the path cost metric that is computed based on a hop count, a message transfer (MTU) size, a link speed, width and other port and link characteristics.